**NumberFormatException In Java:**

**A Java NumberFormatException usually occurs when you try to convert a String to a numeric value, like an int, float, double, long etc.**

**Example 1:**

**public** **class** ConvertStringToInt

{

**private** **static** **int** numberFormatException (String str)

{

**try**

{

// Converting String to Integer

**int** conversionFromStringToInt=Integer.*parseInt*(str);

**return** conversionFromStringToInt;

}

**catch** (NumberFormatException nfe)

{

nfe.printStackTrace();

}

**return** 0;

}

// Main Method

**public** **static** **void** main(String[] args)

{

//String str="NumberFormatException Demo"; //👉 This case will raise NumberFormatException as we are trying to convert String to an Integer.

String str="23";

**int** result=*numberFormatException*(str); //👍🏻 This case will work fine. No NumberFormatException will be raised as we convert integer number to integer.

System.***out***.println(result);

}

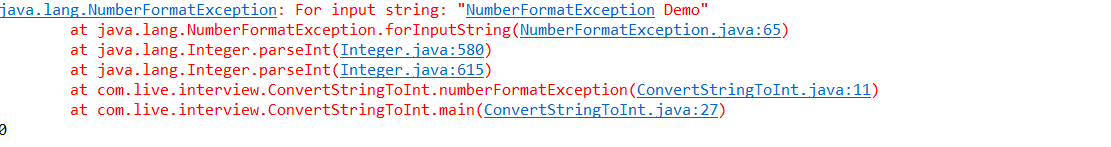
}

**OUTPUT**

**23**

**Example 2:**

In case when string contains alphanumeric character or empty string instead of showing below default message

****

**Can we show the appropriate message.**  I mean if string contains alphanumeric character then display exact message that string contains alphanumeric characterS

**package** com.live.interview;

/\*

\* ✅ NumberFormatException :

NumberFormatException occurs when we try to 🔴convert an string to integer🔴.

So, if string contains numeric value let's say String s1="12345" in that case it won't throw

NumberFormatException but if string contains any string character for instance String s2="Vijay123"

then it will raise NumberFormatException.

👉👉 So this program is developed to address the above mentioned issue,

Can we show the appropriate message. I mean if string contains alphanumeric character then display exact message that string contains alphanumeric character

\*/

**public** **class** ConvertStringToInt

{

**private** **static** **int** numberFormatException(String str)

{

str=str.trim(); // 👍 trim() method is used to remove the white space at the start and end of the string.

str=str.replaceAll("\\s", ""); // Metacharacter //s is used to remove the whitespace

**int** conversionFromStringToInt=0;

**try**

{

// It is a best practice to check if the passed-in string is numeric or not before trying to convert it to integer. If it is integer or contains integer then will convert it to integer else will display appropriate message

**boolean** isNumeric=*checkWhetherStringContainsNumeric*(str);

**if**(isNumeric==**true**)

{

// Converting String to Integer can be done by below 3 ways

//1st Way:

conversionFromStringToInt=Integer.*parseInt*(str);

System.***out***.println("Since string contains numeric digits only we have successfully converted this string to int and here it is:" +conversionFromStringToInt);

//2nd Way :

//conversionFromStringToInt=Integer.valueOf(str);

//3rd Way :

//Integer.format()

}

**else**

{

**boolean** checkString=*checkWhetherItContainsString*(str);

**if**(checkString==**true**)

{

System.***out***.println("Sorry conversion from String to int is not possible as your string contains Alphabets chaacter or either it is null or empty:");

}

}

}

**catch** (NumberFormatException nfe)

{

nfe.printStackTrace();

//Logic added to display the appropriate message

//STEP 1: Converting an int to String to check whether it is null or contain alphanumeric

}

**return** 0;

}

// Method to check whether string is null or empty

**private** **static** **boolean** checkWhetherStringIsEmptyOrNull(String str)

{

**if**(!str.equals("") && str!=**null**)

{

**return** **true**;

}

**else**

**return** **false**;

}

// Method to check if it contains string character

**private** **static** **boolean** checkWhetherItContainsString(String str)

{

**if**(str.matches("^[a-zA-z]\*$") && !str.equals("") && str!=**null**) //👀 👀 !str.equals("") == !str.isEmpty() Menas we can use either !str.equals("") to check empty string or either we can user !str.isEmpty() to check the empty string

{

**return** **true**;

}

**else**

**return** **false**;

}

//Method to check whether it contains numeric value or not.

**private** **static** **boolean** checkWhetherStringContainsNumeric(String str)

{

**if**(str.matches("[0-9.]+") && !str.equals("") && str!=**null**) //👀 👀 !str.equals("") == !str.isEmpty() Menas we can use either !str.equals("") to check empty string or either we can user !str.isEmpty() to check the empty string

{

**return** **true**;

}

**else**

**return** **false**;

}

// Main Method

**public** **static** **void** main(String[] args)

{

String str="NumberFormatException Demo"; //👉 This case will raise NumberFormatException as we are trying to convert String to an Integer.

//String str="23";

**int** result=*numberFormatException*(str); //👍🏻 This case will work fine. No NumberFormatException will be raised as we convert integer number to integer.

System.***out***.println(result);

}

}